Towards Systematic Understanding of Geometric Representations in BIM Standard: An Empirical Data-Driven Approach

Jiansong ZHANG[1]

[1] School of Construction Management Technology, Purdue University, West Lafayette, IN 47907; PH (765) 494-1574; FAX (765) 496-2246; email: jiansong-zhang@purdue.edu

ABSTRACT

The use of Industry foundation classes (IFC) data can facilitate interoperability of building information modeling (BIM) among different applications to alleviate the problems of information missing and inconsistency. By virtue of its goodwill of transparency and openness, IFC data can be opened and viewed in any text editor. But it normally requires a significant amount of effort when manually interpreting IFC data, due to (1) its large number of entities; and (2) the complex connections between one entity and another. On the other hand, the explanations of IFC entities in the IFC schema specifications are difficult to understand or verify. To address such difficulties, in this paper, an empirical data-driven approach is proposed for achieving a systematic understanding of entity definitions in an IFC schema. The approach utilizes IFC data and schema in a synergistic way, to facilitate such systematic understanding. Experimental testing is used to serve as verifications of the understanding and accrue the understanding, along with which byproduct BIM tools will be developed. The proposed approach was tested on understanding entities for geometric representations in the IFC2X3_TC1 schema. Through the experimental testing, systematic understanding of 62 IFC entities were obtained, and a visualization algorithm was developed and implemented based on this understanding.

INTRODUCTION

Building information modeling was created with interoperability in mind. However, BIM interoperability problems have been repeatedly reported (Young et al. 2009). For example, the widely known NIST report (Gallaher et al. 2004) estimated the cost due to the lack of interoperability in the capital facilities industry to be $15.8 billion/year in the US. Solving the interoperability problem requires an agreement on data representation protocols, which led to the development of multiple BIM data standards such as CIMSteel Integration Standards (CIS/2) and industry foundation classes (IFC) (Isikdag et al. 2007). While the CIS/2 defines a data exchange format for structural steel project (NIST 2017), IFC was designed to be a comprehensive data schema covering all grounds for the building and construction industry. IFC is open and neutral, and registered as ISO 16739. By virtue of its goodwill of transparency and openness, IFC data can be opened and viewed in any text editor. However, in spite of the elaborate documentation of IFC schema that is freely available, it normally requires a significant amount of effort when manually interpreting IFC data, due to (1) its large number of entities; and (2) the complex connections between one entity and another. Furthermore, without referencing tangible data samples, the explanations of each individual entity and its attributes appear to be isolated islands that are difficult to be connected with each other. It is also difficult to verify if a reader's understanding of the entity/attributes explanations is accurate. To address this problem, the author proposes an empirical data-driven approach to facilitate understanding and verification of the specifications that explain entities and attributes in an IFC schema. The approach takes advantage of freely available IFC data and builds tools on top of it. A correct understanding of the entities and attributes can be easily verified by observing and analyzing outputs from the tools built, through comparison with existing BIM tools.

BACKGROUND

BIM interoperability and IFC standard

The lack of interoperability is a major barrier to BIM and one of the top issues that need to be addressed to improve the value of BIM (Poirier et al. 2014; Young et al. 2009). The interoperability problem reveals itself in multiple dimensions, such as technological, procedural, organizational, and contextual dimensions which are interrelated with each other (Poirier et al. 2014). The importance of this BIM interoperability problem led to many research efforts, among which a heavy trend towards technological discussions around IFC was observed (Poirier et al. 2014).

The IFC standard is deemed a promising open data standard for building and construction industry data. It was defined using the Standard for Exchange of Product model data (STEP) language that was registered as ISO 10303; and IFC was registered as ISO 16739 (buildingSMART 2017). The IFC standard consists of over 600 entity definitions and over 300 property sets. Geometric representation usually takes a large portion of entities in an IFC data. For example, in the "Duplex Apartment" IFC data published by buildingSMARTalliance of the National Institute of Building Sciences (will be referred to as Duplex Apartment Data hereafter) (East 2013), more than 71.6% (27866* out of 38898) of the entities were directly used for representing geometric information. A building element in IFC may have multiple geometric representations such as "Body" and "Axis" (Geiger et al. 2015). Correspondingly, different types of geometric representations utilize different types of geometric models. For example, "Body" geometric representation uses "SolidModel", whereas "Axis" geometric representation uses "Curve2D." Within "SolidModel" three major subtypes are defined, including "Swept Solid," "Boolean Results," and "Brep Bodies" (buildingSMART 2014a). IFC data using certain geometric models are straightforward to understand such as "Curve2D." For example, Fig. 1 shows a "Curve2D" type of geometric representation in the Duplex Apartment Data. It can be easily understood as a "Curve2D" represented by an IfcPolyline**, which is further represented by two IfcCartesianPoint instances with (X, Y) coordinate values of (7.875799999999975, -11.825) and (7.875799999999981, -8.075000000000001), respectively. IFC data using certain geometric models are not easy to understand, such as "Clipping," which is a subtype of "Boolean Results." For example, Figure 2 shows a "Clipping" type of geometric representation in the Duplex Apartment Data. While it is not difficult to see the "Clipping" is represented by the difference between the IfcExtrudedAreaSolid (#24580) and the IfcPolygonalBoundedHalfSpace (#24592), how the geometric details are represented, especially how the IfcPolygonalBoundedHalfSpace is represented, is not readily observable. The conceptual illustration for IfcPolygonalBoundedHalfSpace provided by buildingSMART (buildingSMART 2014b) comes to great help in facilitating such understanding, by clearly laying out the relative positions of IfcPlane (#24589), object placement represented as IfxAxis2Placement3D (#24588), agreement flag (second argument in #24592), and polygonal boundary represented as IfcPolyline (#24586), as well as how they are used to form the "half space." Nevertheless, without reference to sample data,

*counting instances in 31 types of IFC entities for geometric representation, such as ifccartesianpoint, ifcpolyline, and ifcface from the data file "Duplex_A_20110907.ifc"
**for readability, IFC entity names will be represented in CamelCase rather than UPPER CASE

The published version is found in the ASCE Library here: https://ascelibrary.org/doi/abs/10.1061/9780784481264.010
Zhang, J. (2018). "Towards systematic understanding of geometric representations in BIM standard: an empirical data-driven approach." Proc., ASCE Construction Research Congress, ASCE, Reston, VA, 96-105.

doubts can occur. For example, in the explanation of IfcPolygonalBoundedHalfSpace provided by buildingSMART (buildingSMART2014b), the use of agreement flag is explained as “If the agreement flag is TRUE, then the subset is the one the normal points away from. If the agreement flag is FALSE, then the subset is the one the normal points into.” Doubts on this true meaning of agreement flag may arise for two reasons: (1) normally TRUE is used to refer to the positive case (i.e., the direction the normal points into); and (2) the potential ambiguity of “away from” caused by the unbounded nature of a half space, because both the positive case and negative case seem reasonable to say it is the one the normal points away from, considering that in the positive case the half space can be regarded as the departure of the normal. Without testing, such doubts are not easy to get cleared.

```
#8428=IFCCARTESIANPOINT((7.875799999999975,-11.825));
#8429=IFCCARTESIANPOINT((7.875799999999981,-8.075000000000001));
#8430=IFCPOLYLINE((#8428,#8429));
#8431=IFCSHAPEREPRESENTATION(#27,'WalkingLine','Curve2D',(#8430));
```

Figure 1. Sample geometric data using “Curve2D”

```
#24580=IFCEXTRUDEDAREASOLID(#24578,#24579,#9,2.795000000000196);
#24581=IFCPRESENTATIONSTYLEASSIGNMENT((#3792));
#24582=IFCSTYLEDITEM(#24580,(#24581),$);
#24583=IFCCARTESIANPOINT((3.707999999999989,-0.));
#24584=IFCCARTESIANPOINT((3.707999999999987,0.1520000000000001));
#24585=IFCCARTESIANPOINT((-0.,0.1520000000000001));
#24586=IFCPOLYLINE((#4,#24583,#24584,#24585,#4));
#24587=IFCCARTESIANPOINT((0.,-0.07600000000000005,2.795000000000196));
#24588=IFCAXIS2PLACEMENT3D(#24587,#10,#7);
#24589=IFCPLANE(#24588);
#24590=IFCCARTESIANPOINT((0.,-0.07600000000000005,2.795000000000196));
#24591=IFCAXIS2PLACEMENT3D(#24590,$,$);
#24592=IFCPOLYGONALBOUNDEDHALFSPACE(#24589,.T.,#24591,#24586);
#24593=IFCBOOLEANCLIPPINGRESULT(.DIFFERENCE.,#24580,#24592);
#24594=IFCSHAPEREPRESENTATION(#27,'Body','Clipping',(#24593));
```

Figure 2. Sample geometric data using “Clipping”

BIM visualization

Visualization of the geometry is an important part of almost all BIM software that support IFC. Corresponding to the main objective of each BIM software, the visualization comes at different levels of details and effects. Figure 3 shows the visualization results of the same portion of the Duplex Apartment Data by different BIM software such as BIM Vision, Constructivity, DDS CAD Viewer, and Solibri Model Viewer, using their respective default settings.

All visualization effects come from the geometric data in a BIM model, a systematic understanding of geometric data is therefore not only important for creating the needed visualization for a BIM tool development, but also important for many BIM research that relies on the geometric data as part of the information need. Moreover, such understanding is critical for the intended interoperability of BIM.

The published version is found in the ASCE Library here: https://ascelibrary.org/doi/abs/10.1061/9780784481264.010
Zhang, J. (2018). "Towards systematic understanding of geometric representations in BIM standard: an empirical data-driven approach." Proc., ASCE Construction Research Congress, ASCE, Reston, VA, 96-105.

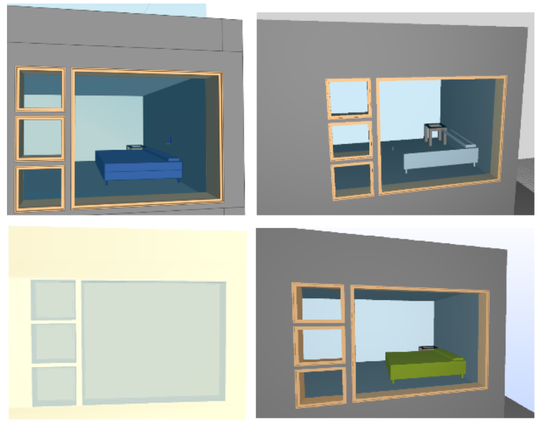

Figure 3. Sample visualization in different BIM software

IFC data processing

By virtue of the openness of IFC, many programming resources for accessing and processing IFC data are available, such as IFCToolboX (Eurostep 2002), ifcplusplus (ifcPlusPlus 2015), Java Toolbox IFC2x3/IFC4 (IFC Tools Project 2013), Open IFC tools (Open IFC Tools 2010), and JSDAI (LKSoftWare GmbH 2017). The rationale behind most such programming resources is enabling direct access of IFC entities by compiling the entity definitions from the IFC schemas into usable classes in the corresponding programming language. While some resources only provide the compiled classes, a family of tools implementing the Standard Data Access Interface (SDAI) also provide utilities for compiling IFC schemas. IFC schemas are defined in the STEP language, and STEP application protocols are defined using EXPRESS language. SDAI is the application programming interface (API) to EXPRESS defined data therefore inherently suitable for processing IFC data. In fact, general SDAI operations are registered as international standard ISO 10303-22 (STEP Tools 2017). SDAI has been implemented in C, C++, and Java programming languages and is under development for Python programming language (STEP Tools 2017; STEPcode 2017). JSDAI is the implementation of SDAI in Java programming language and was used in several earlier studies (Zhang and El-Gohary 2015).

PROPOSED APPROACH

The author proposes an empirical data-driven approach to achieving systematic understanding of specifications that explain entities and attributes in an IFC schema. As shown in Figure 4, the approach takes (1) IFC data, and (2) due diligence in experimental testing, to produce systematic understanding of the IFC schema, while at the same time byproduct in the form of BIM utilities or tools is generated. Therefore, the author named it DSD approach because it combines Data, Schema, and due Diligence. The byproduct is closely tied with the systematic understanding, which can serve as a verification of the understanding as well as records for others to repeat such verification. The DSD approach can also be interpreted as Data-drive Software Development approach from the perspective of the generated byproduct software. The experimental testing is conducted in a bootstrapping manner: it starts with a small data sample to gain understanding of a small portion of entities in the IFC schema, and keeps expanding itself by iteratively testing more and more entity definitions in the IFC schema and accumulating knowledge and understanding correspondingly. Because entities in the IFC data are interconnected, the understanding of different parts of the IFC schema also serves as verifications of each other. In this iterative testing and accumulation process, earlier understanding may be adjusted or in some cases overturned in later steps, which unsurprisingly mirrors a general phenomenon in knowledge discovery in any scientific domain.

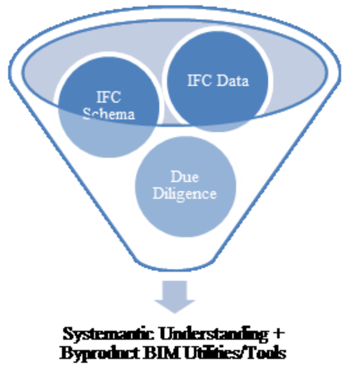

Figure 4. Proposed approach for systematic understanding of IFC standard

EXPERIMENT

To test the proposed approach, an experiment was conducted on understanding geometric representations of the IFC2X3_TC1 schema (buildingSMART 2007). The experiment started with a simple bridge model with one deck and four piers (of cone frustum shape) created by Mandava and Zhang (2016). To support the visualization, Java 3D (Java3d 2012) was used. JSDAI (LKSoftWare GmbH 2017) was used to access entity information in an IFC model. Figure 5 shows partial data for representing the geometry of one pier in the bridge. For a holistic understanding of the Cartesian points used in this geometric representation, these points were manually extracted and visualized in a dynamic gaming environment which allows a constant change of perspectives in a first-person view (Figure 6). It was observed that sixteen Cartesian points were used to represent the top surface (of circle shape) of the cone frustum and sixteen Cartesian points were used to represent the bottom surface (of circle shape) of the cone frustum. Based on this observation, further analysis was conducted where each entity used in this geometric representation was looked up in the specifications provided by buildingSMART (buildingSMART 2007). The following understanding was obtained: a IfcBuildingElementProxy has nine attributes among which the seventh attribute is its representation, the representation can use IfcProductDefinitionShape; each IfcProductDefinitionShape has three attributes among which the third attribute is a list of representations [e.g., (#41, #42)], a representation can use IfcShapeRepresentation; each IfcShapeRepresentation has four attributes among which the fourth attribute is a set of representation item(s), a representation item can use IfcFacetedBrep (e.g., #40); each IfcFacetedBrep has one attribute which is the outer boundary, an outer boundary can use IfcClosedShell (e.g., #307); each IfcClosedShell has one attribute which is a set of faces, a face can use IfcFace (e.g., #393); each IfcFace has one attribute which is a set of bounds, a bound can use IfcFaceOuterBound (e.g., #392); each IfcFaceOuterBound has two attributes among which the first attribute is the bound, the bound can use IfcPolyLoop (e.g., #391); each IfcPolyLoop has one attribute which is a list of Cartesian points depicting the vertices of the poly loop (e.g., #327); each IfcCartesianPoint has one attribute which is a list of its xyz coordinates. To verify this understanding, a visualization algorithm was developed correspondingly to visualize the geometry of the bridge pier which was programmed in Java 3D. The visualization results using the algorithm in comparison with a commercial tool are shown in Figure 7. This verifies the understanding of

The published version is found in the ASCE Library here: https://ascelibrary.org/doi/abs/10.1061/9780784481264.010
Zhang, J. (2018). "Towards systematic understanding of geometric representations in BIM standard: an empirical data-driven approach." Proc., ASCE Construction Research Congress, ASCE, Reston, VA, 96-105.

the IFC entities described above. Figure 8 shows the flowchart of the main processes in the algorithm for visualization.

```
#33=IFCPRODUCTDEFINITIONSHAPE($,$,(#41,#42));
#40=IFCFACETEDBREP(#307);
#41=IFCSHAPEREPRESENTATION(#26,'Body','Brep',(#40));
#42=IFCSHAPEREPRESENTATION(#27,'Box','BoundingBox',(#29));
#43=IFCBUILDINGELEMENTPROXY('01mbfVTizDwvOpPsA2JtyA',#16,'','0, one Pier.dgn, Default:672',$,#47,#33,$,$);
#307=IFCCLOSEDSHELL((#326,#331,#335,#339,#343,#347,#351,#355,#359,#363,#367,#371,#375,#379,#383,#387,#390,#393));
#327=IFCCARTESIANPOINT((6.673181,-0.476835,3.250083));
#391=IFCPOLYLOOP((#384,#380,#376,#372,#368,#364,#360,#356,#352,#348,#344,#340,#336,#332,#328,#327));
#392=IFCFACEOUTERBOUND(#391,.T.);
#393=IFCFACE((#392));
```

Figure 5. Partial data representing geometry of a simple bridge pier

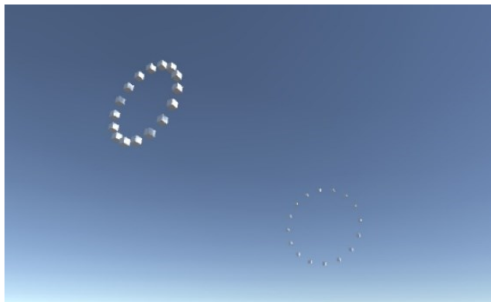

Figure 6. Observation of Cartesian points in a game engine

The visualization algorithm was directly applied to the whole bridge (i.e., one deck and four piers). A bird eye view of the visualization results is shown in Figure 9. Observation found that the geometries of the five components appeared correct, but the relative positions between the components appeared problematic. An analysis was conducted which showed the dislocation problem was due to the lack of implementation of "local placement" information which was provided by IfcLocalPlacement.

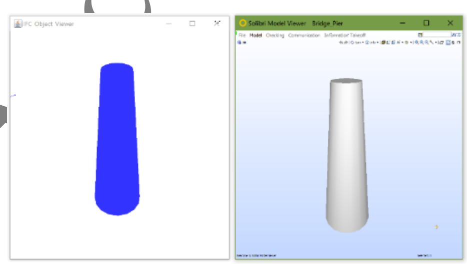

Figure 7. Visualization of a cone frustum shape: verification through comparative observation

The published version is found in the ASCE Library here: https://ascelibrary.org/doi/abs/10.1061/9780784481264.010
Zhang, J. (2018). "Towards systematic understanding of geometric representations in BIM standard: an empirical data-driven approach." Proc., ASCE Construction Research Congress, ASCE, Reston, VA, 96-105.

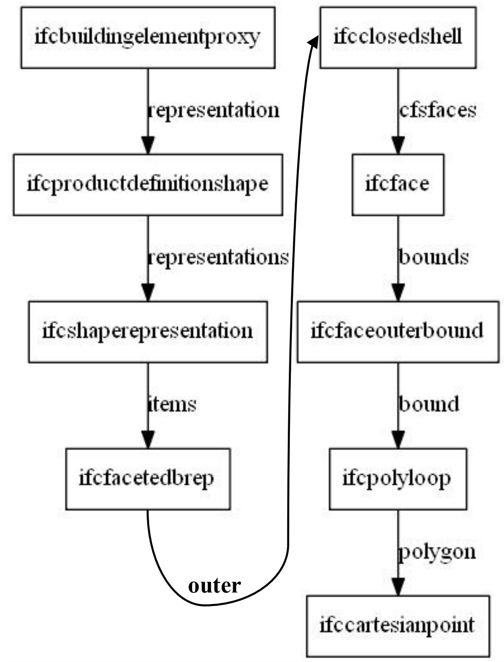

Figure 8. Flowchart of visualization algorithm

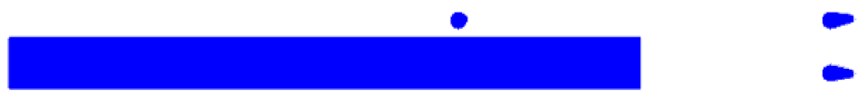

Figure 9. Bird's-eye view visualization of a deck and four piers

Each IfcLocalPlacement has two attributes: the first attribute is "PlacementRelTo" which defines a reference for the local placement; the second attribute is "RelativePlacement" which defines the translation and rotation parameters for the transformations from the reference. The "RelativePlacement" is usually defined by an IfcAxis2Placement3D. An algorithm for addressing the transformation was developed which extracts translation and rotation parameters from an IfcAxis2Placement3D and applied them to the reference. When the reference is another IfcLocalPlacement, this process is iterated again. The iteration ends when the reference is empty (i.e., using default world coordinate). After incorporating this transformation algorithm into the visualization algorithm, the visualization result of the bridge appeared correct (Figure 10). The experiment was continued on the Duplex Apartment Data, where understanding of more entities were obtained and verified using testing algorithm developed based on the understanding.

Figure 10. Visualization of the simple bridge

The published version is found in the ASCE Library here: https://ascelibrary.org/doi/abs/10.1061/9780784481264.010
Zhang, J. (2018). "Towards systematic understanding of geometric representations in BIM standard: an empirical data-driven approach." Proc., ASCE Construction Research Congress, ASCE, Reston, VA, 96-105.

EXPERIMENTAL RESULTS AND ANALYSIS

Through the experiment, systematic understanding of 62 entities in IFC related to geometric representations were obtained (Figure 11).

```
['ifccartesianpoint', 'ifcproductdefinitionshape', 'ifcshaperepresentation', 'ifcpolyline', 'ifcextru
dedareasolid', 'ifcgeometricset', 'ifcbooleanclippingresult', 'ifcfacebasedsurfacemodel', 'ifcmappedi
tem', 'ifcarbitraryprofiledefwithvoids', 'ifcrectangleprofiledef', 'ifcarbitraryclosedprofiledef', 'i
fccircleprofiledef', 'ifcarbitraryopenprofiledef', 'ifcsurfaceoflinearextrusion', 'ifccompositecurve'
, 'ifccompositecurvesegment', 'ifcfacebound', 'ifcpolyloop', 'ifcfaceouterbound', 'ifcface', 'ifcconn
ectedfaceset', 'ifcpolygonalboundedhalfspace', 'ifcplane', 'ifccurveboundedplane', 'ifctrimmedcurve',
'ifcaxis', 'ifcdirection', 'ifcgeometricrepresentationsubcontext', 'ifcgeometricrepresentationcontext
', 'ifcbuildingelementproxy', 'ifcstairflight', 'ifcrailing', 'ifcmember', 'ifcfooting', 'ifcwallstan
dardcase', 'ifcwall', 'ifcfurnishingelement', 'ifcslab', 'ifccovering', 'ifcdoor', 'ifcwindow', 'ifco
peningelement', 'ifcspace', 'ifcmaterial', 'ifcrelassociatesmaterial', 'ifcmateriallayersetusage', 'i
fcmateriallayerset', 'ifcmateriallayer', 'ifcmaterialdefinitionrepresentation', 'ifcstyledrepresentat
ion', 'ifcstyleditem', 'ifcpresentationstyleassignment', 'ifcsurfacestyle', 'ifcsurfacestylerendering
', 'ifcfillareastyle', 'ifccartesiantransformationoperator', 'ifcrepresentationmap', 'ifccolourrgb',
'ifcfillareastylehatching', 'ifccurvestyle', 'ifcrelvoidselement']
```

Figure 11. IFC entities covered in the systematic understanding

Based on the understanding the visualization algorithm was extended which successfully visualized the walls, windows, doors, furnishing elements, stairs and railings, footings, slabs, and coverings in the Duplex Apartment Data. Figure 12 shows the comparison of the visualization results with results from a commercial IFC viewer. Such comparison can straightforwardly verify if the understanding and corresponding processing algorithms of the IFC entities were correct. In the visualization, certain elements (e.g., furnishing elements) were using arbitrary colors because no color assignment was found in the data. While there are small differences in terms of texture and lighting usage, it was not the focus of this study as the author focused on geometric representations. Nevertheless, color data were used when detected in the IFC data, such as the color of the exterior walls and roof.

Figure 12. Visualization of the Duplex Apartment Data: verification through comparative observation

CONCLUSION

In this study, an empirical data-driven approach was proposed to support systematic understanding of IFC entity definitions. The approach takes (1) IFC data, and (2) due diligence in experimental testing, to produce systematic understanding of the IFC schema, while byproduct is generated in the form of BIM utilities or tools. The systematic understanding is obtained through iterative experimental testing in a bootstrap manner. To test the proposed approach, an experiment was conducted which started with a simple bridge IFC model and expanded to the open Duplex Apartment Data model provided by the National Institute of Building Sciences. Systematic understanding of 62 IFC entities were obtained and a byproduct of visualization algorithm was created. The visualization algorithm successfully processed and visualized the bridge model and the Duplex Apartment Data model. Through comparison of the visualization results with results from a commercial BIM viewer, the systematic understanding was verified. Such systematic understanding of IFC entities is important not only from the practical BIM application development perspective, but also for supporting future BIM research in many areas such as construction operation automation, where geometric information of BIM objects is critical.

ACKNOWLEDGMENTS

The author would like to thank the National Science Foundation (NSF). This material is based on work supported by the NSF under Grant No. 1745374. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the NSF.

REFERENCES

buildingSMART. (2007). “IFC2x Edition 3 Technical Corrigendum 1” <http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/> (Aug. 2, 2017).

buildingSMART. (2014a). “IfcShapeRepresentation.” <http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/ifcrepresentationresource/lexical/ifcshaperepresentation.htm> (Aug. 1, 2017).

buildingSMART. (2014b). “IfcPolygonalBoundedHalfSpace.” <http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/ifcgeometricmodelresource/lexical/ifcpolygonalboundedhalfspace.htm> (Aug. 1, 2017).

buildingSMART. (2017). “IFC overview summary.” <http://www.buildingsmart-tech.org/specifications/ifc-overview> (Jul. 31, 2017).

East, E.W. (2013). “Common building information model files and tools.” < http://www.nibs.org/?page=bsa_commonbimfiles&hhSearchTerms=%22common+and+BIM+and+file%22> (Jun. 27, 2014).

Eurostep. (2002). “IFCToolboX: Extended tool support for IFC implementations.” <http://cic.vtt.fi/vera/projects/e_ifctoolbox.htm> (Aug. 2, 2017).

Gallaher, M.P., O’connor, A.C., Dettbarn JR., J.L. & Gilday, L.T. (2004). “Cost analysis of inadequate interoperability in the U.S. capital facilities industry.” *NIST Publication GCR* 04-867.

Geiger, A., Benner J., and Haefele, K.H. (2015). “Generalization of 3D IFC building models.” M. Breunig et al. (eds.), 3D Geoinformation Science, Lecture Notes in Geoinformation and Cartography, 19-35.

ifcPlusPlus. (2016). <https://github.com/berndhahnebach/IfcPlusPlus> (Aug. 2, 2017).

IFC Tools Project. (2013). <http://www.ifctoolsproject.com/> (Aug. 2, 2017).

Isikdag, U., Aouad, G., Underwood, J, and Wu, S. (2007). "Building information models: a review on storage and exchange mechanisms." Proc., 24th W78 Conf.& 5th ITCEDU Workshop & 14th EG-ICE Workshop, Bringing ITC Knowledge to Work, International Council for Research and Innovation in Building and Construction (CIB), Rotterdam, Netherlands, 135-144.

Java3d. (2012). "Java 3D Tutorial." <http://www.java3d.org/> (Aug. 2, 2017).

LKSoftWare GmbH. (2017). "JSDAI express compiler." <http://www.jsdai.net/core/express-compiler> (Aug. 2, 2017).

Mandava, B., and Zhang, J. (2016). "A new automated quantity takeoff method for BIM-based bridge designs." Proc., CIB W78, Conseil International du Bâtiment (CIB), Rotterdam, The Netherlands.

NIST. (2017). "SteelVis (aka CIS/2 Viewer)." <https://www.nist.gov/services-resources/software/steelvis-aka-cis2-viewer>. (Aug. 3, 2017).

Open IFC Tools. (2010). <http://www.openifctools.org> (Aug. 2, 2017).

Poirier, E.A., Forgues, D., and Staub-French, S. (2014). "Dimensions of interoperability in the AEC industry." Proc., 2014 ASCE Construction Research Congress (CRC), Atlanta, Georgia, May 19-21, 2014, 1987-1994.

STEPcode. (2017). "STEPcode v0.8" <https://github.com/stepcode/stepcode/wiki/python-generator> (Aug. 2, 2017).

STEP Tools. (2017). "The STEP standard - ISO 10303." <https://www.steptools.com/stds/step/sdai.html> (Aug. 2, 2017).

Young, N.W.Jr., Jones, S.A., Bernstein, H.M., and Gudgel, J.E. (2009). "The business value of BIM: Getting building information modeling to the bottom line." < http://fiatech.org/images/stories/research/2009_BIM_SmartMarket_Report.pdf> (Jul. 31, 2017).

Zhang, J., and El-Gohary, N. (2015). "Automated extraction of information from building information models into a semantic logic-based representation." Proc., 2015 ASCE Intl. Workshop on Comput. in Civ. Eng., ASCE, Reston, VA, 173-180.